### **🕹️ Game: “Escape the Haunted Maze”**

🔹 **Problem Statement:**  
You are trapped in a **haunted maze** 🏚️, and ghosts are chasing you! 👻 Your goal is to **escape the maze** by reaching the exit E without being caught by a ghost G.  
- The maze is represented as a **grid** (NxM matrix).  
- P represents the **player** (you).  
- G represents **ghosts** that move **towards** the player each turn.  
- E represents the **exit** (you win if you reach it).  
- # represents **walls** (you can’t pass through them).  
- . represents **empty spaces** where you can move.  
- If a ghost reaches you, **you lose**! 💀

✅ The player can move **UP (W), DOWN (S), LEFT (A), RIGHT (D)**.  
✅ Ghosts **automatically move** toward the player after each turn.  
✅ The player **wins** if they reach E.  
✅ The player **loses** if a ghost catches them.

### **🔧 Stubbed Code**

import random  
import time  
  
# Define the maze grid  
maze = [  
 ["#", "#", "#", "#", "#", "#", "#", "#"],  
 ["#", "P", ".", ".", "#", ".", "E", "#"],  
 ["#", "#", "#", ".", "#", ".", "#", "#"],  
 ["#", ".", "G", ".", ".", ".", ".", "#"],  
 ["#", "#", "#", "#", "#", "#", "#", "#"]  
]  
  
# Function to display the maze  
def print\_maze():  
 for row in maze:  
 print(" ".join(row))  
 print("\n")  
  
# Function to find the position of an element (P, G, E)  
def find\_position(symbol):  
 for i in range(len(maze)):  
 for j in range(len(maze[i])):  
 if maze[i][j] == symbol:  
 return (i, j)  
 return None  
  
# Function to move the player  
def move\_player(direction):  
 global maze  
 px, py = find\_position("P")  
   
 # Calculate new position based on input  
 new\_x, new\_y = px, py  
 if direction == "W": new\_x -= 1 # Up  
 elif direction == "S": new\_x += 1 # Down  
 elif direction == "A": new\_y -= 1 # Left  
 elif direction == "D": new\_y += 1 # Right  
   
 # TODO: Handle wall collisions  
 # TODO: Handle reaching the exit  
 # TODO: Handle valid movement updates  
  
# Function to move the ghosts towards the player  
def move\_ghosts():  
 global maze  
 px, py = find\_position("P")  
 gx, gy = find\_position("G")  
  
 # TODO: Implement logic to move ghost closer to player  
  
# Game loop  
def game\_loop():  
 print("🎃 Welcome to the Haunted Maze! Escape before the ghosts catch you! 👻")  
 print\_maze()  
   
 while True:  
 direction = input("Move (W/A/S/D): ").upper()  
 if direction not in ["W", "A", "S", "D"]:  
 print("Invalid move! Use W (up), A (left), S (down), D (right)")  
 continue  
   
 move\_player(direction) # Move player  
 move\_ghosts() # Move ghosts  
 print\_maze()  
   
 # TODO: Check win condition  
 # TODO: Check lose condition  
  
game\_loop()

### **📌 Your Task**

1. **Complete the move\_player() function**
   * Ensure the player can’t move through walls.
   * If the player reaches E, print "You escaped! 🎉" and exit the game.
2. **Complete the move\_ghosts() function**
   * Make ghosts **move one step closer** to the player after each turn.
   * If a ghost reaches the player, print "You got caught! 💀" and exit the game.
3. **Implement win/loss conditions**
   * If the player reaches E, they **win**.
   * If a ghost reaches P, they **lose**.

### **🎯 Expected Output Example**

🎃 Welcome to the Haunted Maze! Escape before the ghosts catch you! 👻  
# # # # # # # #  
# P . . # . E #  
# # # . # . # #  
# . G . . . . #  
# # # # # # # #  
  
Move (W/A/S/D): D  
  
# # # # # # # #  
# . P . # . E #  
# # # . # . # #  
# . G . . . . #  
# # # # # # # #  
  
Move (W/A/S/D): S  
  
# # # # # # # #  
# . . . # . E #  
# # # P # . # #  
# . G . . . . #  
# # # # # # # #  
  
Move (W/A/S/D): D  
  
🎉 You escaped! 🎉

### **⏳ Time Limit: 15 Minutes**

💡 **Hint:**  
- Use find\_position("P") to get the player’s coordinates.  
- Use find\_position("G") to get the ghost’s coordinates.  
- Use simple movement logic for ghosts (move in the shortest path).